QUIZ 6

COMP9021 PRINCIPLES OF PROGRAMMING

Sample outputs

```
$ python3
...
>>> from quiz_6 import *
>>> Permutation('No way!')
...
quiz_6.PermutationError: Cannot generate permutation from these arguments
>>> Permutation([3, 2, 1])
...
quiz_6.PermutationError: Cannot generate permutation from these arguments
>>> Permutation(3, 2, 1, length = 4)
...
quiz_6.PermutationError: Cannot generate permutation from these arguments
>>> Permutation(length = -1)
...
quiz_6.PermutationError: Cannot generate permutation from these arguments
>>> Permutation(2, 1, 0)
...
quiz_6.PermutationError: Cannot generate permutation from these arguments
>>> p = Permutation(3, 2, 1)
>>> p
Permutation(3, 2, 1)
>>> p = Permutation(5, 2, 1, 3, 4, length = 5)
>>> p
Permutation(5, 2, 1, 3, 4)
>>> p = Permutation()
>>> p
Permutation()
>>> print(p)
()
>>> len(p)
0
>>> p.nb_of_cycles
0
>>> p = Permutation(length = 4)
>>> p
Permutation(1, 2, 3, 4)
>>> print(p)
(1)(2)(3)(4)
>>> len(p)
4
>>> p.nb_of_cycles
4
```

Date: Session 2, 2017.

```
>>> p = Permutation(2, 3, 4, 5, 1)
>>> p
Permutation(2, 3, 4, 5, 1)
>>> print(p)
(5 1 2 3 4)
>>> len(p)
5
>>> p.nb_of_cycles
1
>>> q = p.inverse()
>>> p
Permutation(2, 3, 4, 5, 1)
>>> q
Permutation(5, 1, 2, 3, 4)
>>> print(q)
(5 4 3 2 1)
>>> len(q)
5
>>> q.nb_of_cycles
1
>>> p = Permutation(2, 5, 4, 3, 1, length = 5)
>>> p
Permutation(2, 5, 4, 3, 1)
>>> print(p)
(4 3)(5 1 2)
>>> len(p)
5
>>> p.nb_of_cycles
2
>>> q = p.inverse()
>>> p
Permutation(2, 5, 4, 3, 1)
>>> q
Permutation(5, 1, 4, 3, 2)
>>> print(q)
(4 3)(5 2 1)
>>> len(q)
5
>>> q.nb_of_cycles
2
```

```
>>> Permutation() * Permutation(1)
...
quiz_6.PermutationError: Cannot compose permutations of different lengths
>>> Permutation(1, 2, 3) * Permutation(length = 2)
...
quiz_6.PermutationError: Cannot compose permutations of different lengths
>>> p1 = Permutation(5, 4, 3, 2, 1)
>>> p1
Permutation(5, 4, 3, 2, 1)
>>> print(p1)
(3)(4 2)(5 1)
>>> p2 = Permutation(2, 4, 1, 5, 3)
>>> p2
Permutation(2, 4, 1, 5, 3)
>>> print(p2)
(5 3 1 2 4)
>>> q = p1 * p2
>>> p1
Permutation(5, 4, 3, 2, 1)
>>> p2
Permutation(2, 4, 1, 5, 3)
>>> q
Permutation(3, 5, 1, 4, 2)
>>> print(q)
(3 1)(4)(5 2)
>>> p2 *= p1
>>> p1
Permutation(5, 4, 3, 2, 1)
>>> p2
Permutation(4, 2, 5, 1, 3)
>>> print(p2)
(2)(4 1)(5 3)
```